LA-UR-03-2655

Cite Details

B. Wohlberg and C. Brislawn, "Reversible integer-to-integer transforms and symmetric extension of even-length filter banks", *Proceedings of Visual Communication and Image Processing (VCIP) 2003*, Lugano, Switzerland, pp. 1709-1714, July 2003

SPIE Copyright Notice

Reversible integer-to-integer transforms and symmetric extension of even-length filter banks

Brendt Wohlberg and Christopher Brislawn

Los Alamos National Laboratory, Los Alamos, NM 87545, USA

ABSTRACT

The recent JPEG2000 image coding standard includes a lossless coding mode based on reversible integer to integer filter banks, which are constructed by inserting rounding operations into the filter bank lifting factorisation. The baseline (Part 1) of the JPEG2000 standard supports a single reversible filter bank, the finite length input to which is symmetrically extended to avoid difficulties at the boundaries. While designing support for arbitrary filter banks for Part 2 of the standard, we discovered that reversibility is not always possible for even length integer to integer filter banks combined with symmetric pre-extension.

Keywords: Lossless coding, JPEG2000, integer transform, wavelet transform, lifting, filter bank, reversible transform

1. INTRODUCTION

The lossy coding mode of the recent JPEG2000 standard consists of the following main stages: a decorrelating wavelet transform, quantisation of the transform coefficients, and entropy coding of the bit planes of the quantiser indices. In addition to its role in improving rate-distortion performance, the wavelet transform plays a critical role in enabling the resolution scalability properties of the JPEG2000 bitstream — it is therefore desirable that the lossless coding mode also include a wavelet transform. This is enabled by the construction of *reversible* integer to integer wavelet transforms, which are combined with bit plane entropy coding (quantisation is obviously not appropriate) to form the lossless coding mode. The basic building blocks of these wavelet transforms are two-channel filter banks.

2. TWO-CHANNEL FILTER BANKS

A two-channel filter bank[1] consists of a pair of lowpass and highpass analysis filters (H_0 and H_1), and a pair of lowpass and highpass synthesis filters (G_0 and G_1), as depicted in Figure 1. The output of the analysis stage is obtained by splitting the input into two channels, convolving each channel with the corresponding filter, and downsampling each channel by a factor of two. The process is reversed on synthesis, upsampling each channel by a factor of two before convolution and summation of the the channels. (Following standard practice, we denote signals and filters by their z-transforms, so that convolution is expressed as multiplication. A non-causal finite impulse response filter has a z-transform which is a Laurent polynomial in z — that is, a polynomial in both positive and negative powers of z.) Such a structure is a *perfect reconstruction filter bank* if the filters are chosen so that the output of the synthesis stage is equal to the input to the analysis stage. The standard Discrete Wavelet Transform (Mallat decomposition) is constructed by cascading such two-channel filter banks on the lowpass channel, as depicted in Figure 2.

Direct implementation of a filter bank as illustrated in Figure 1 is inefficient, since half of the samples computed by the analysis convolution are discarded by the downsampling operations, and similar inefficiencies exist on the synthesis side. In the *polyphase* representation, the equivalent of filtering followed by downsampling (a similar identity exists for upsampling followed by filtering) is obtained by summing the contributions from convolving the even subsequence of the input with the even subsequence of the filter, and the odd subsequence of the input

Author email: brendt@lanl.gov, brislawn@lanl.gov.

Los Alamos National Laboratory is operated by the University of California for the U. S. Department of Energy under contract W-7405-ENG-36. The work of both authors was supported in part by the DOE Applied Mathematical Sciences Program, contract KC-07-01-01.

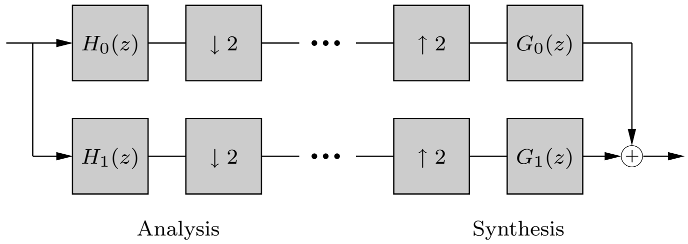

Figure 1. Direct form of a two-channel filter bank. On the analysis side, filtering is followed by downsampling on each channel, while on the synthesis side, filtering is preceded by upsampling on each channel.

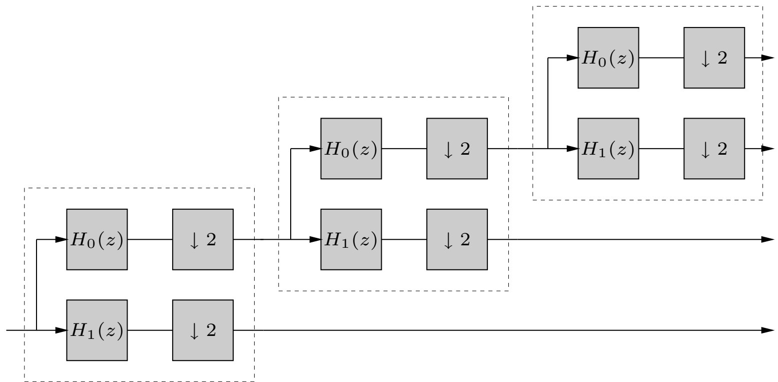

Figure 2. Construction of a three-level wavelet transform from a two-channel filter bank.

with the odd subsequence of the filter[1].[2] In extending this representation to the entire filter bank (see Figure 3), it is convenient to define the polyphase vector form of input $X(z)$ as the vector

$$\mathbf{X}(z) = \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix},$$

having as its components the even and odd subsequences $X_0(z)$ and $X_1(z)$ of $X(z)$, defined so that

$$X(z) = X_0(z^2) + z^{-1}X_1(z^2).$$

The analysis and synthesis operations (see Figure 3) may be expressed as

$$\begin{bmatrix} X_L(z) \\ X_H(z) \end{bmatrix} = \mathbf{H}_a(z) \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix} \qquad \begin{bmatrix} X'_0(z) \\ X'_1(z) \end{bmatrix} = \mathbf{G}_s(z) \begin{bmatrix} X'_L(z) \\ X'_H(z) \end{bmatrix}$$

by defining the analysis and synthesis polyphase matrices

$$\mathbf{H}_a(z) = \begin{bmatrix} H_{a_{00}}(z) & H_{a_{01}}(z) \\ H_{a_{10}}(z) & H_{a_{11}}(z) \end{bmatrix} \qquad \mathbf{G}_s(z) = \begin{bmatrix} G_{s_{00}}(z) & G_{s_{01}}(z) \\ G_{s_{10}}(z) & G_{s_{11}}(z) \end{bmatrix}.$$

These matrices have components consisting of the even and odd subsequences of the lowpass and highpass filters of the analysis and synthesis stages respectively, defined such that

$$\begin{aligned} H_k(z) &= H_{a_{k0}}(z^2) + zH_{a_{k1}}(z^2) \quad k \in \{0,1\} \\ G_k(z) &= G_{s_{0k}}(z^2) + z^{-1}G_{s_{1k}}(z^2) \quad k \in \{0,1\}. \end{aligned}$$

Within this representation, the perfect reconstruction condition is simply $\mathbf{G}_s\mathbf{H}_a = I$.

Note that we utilise the *polyphase with advance* form of polyphase decomposition, which, while differing from the more common delay form used in the filter bank literature[1],[2] is compatible with the conventions established in the JPEG2000 standard (a more detailed discussion of this issue is presented in a companion paper[3]), which was based on the convention adopted in the original paper of Daubechies and Sweldens.[4]

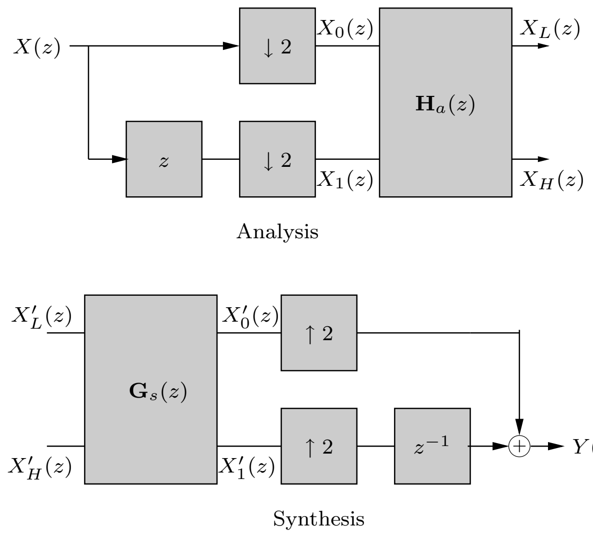

Figure 3. Polyphase form of a two-channel filter bank.

It is possible to show that any polyphase matrix may be factorised[4] into the product of a diagonal matrix and alternating upper and lower diagonal matrices, as in

$$\mathbf{H}_a(z) = \begin{bmatrix} \frac{1}{K} & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} 1 & 0 \\ S_{N_{LS}-1}(z) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & 0 \\ S_1(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & S_0(z) \\ 0 & 1 \end{bmatrix},$$

where the $S_k(z)$ are Laurent polynomials and K is some scalar. This factorisation corresponds to the *lifting* structure illustrated in Figure 4, in which

$$\mathbf{H}_a(z) = \begin{bmatrix} \frac{1}{K} & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} 1 & 0 \\ S_1(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & S_0(z) \\ 0 & 1 \end{bmatrix}$$

and

$$\mathbf{G}_s(z) = \begin{bmatrix} 1 & -S_0(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -S_1(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix}.$$

Some examples of lifting factorisations for common filter banks are presented in Table 1. These factorisations offer a number of advantages, including *in place* implementations and low computational complexity.[4]

3. INTEGER TRANSFORMS

Integer to integer filter banks[5] may be constructed by inserting a rounding operation into each step of a lifting implementation. Viewed as a transform on an input of infinite extent, it is clear from the flow diagram in Figure 5 that any choice of rounding operation $R(\cdot)$ preserves reversibility, since exactly the same branch is subtracted at each step of the synthesis as was added at the corresponding step of the analysis. Note that, for reasons of implementational efficiency, it is also important to use lifting factorisations for which all of the filter coefficients are dyadic rationals (as is the case for all of the filter banks listed in Table 1) so that multiplications of samples by filter coefficients may be implemented as integer multiplications followed by bit-shifts.

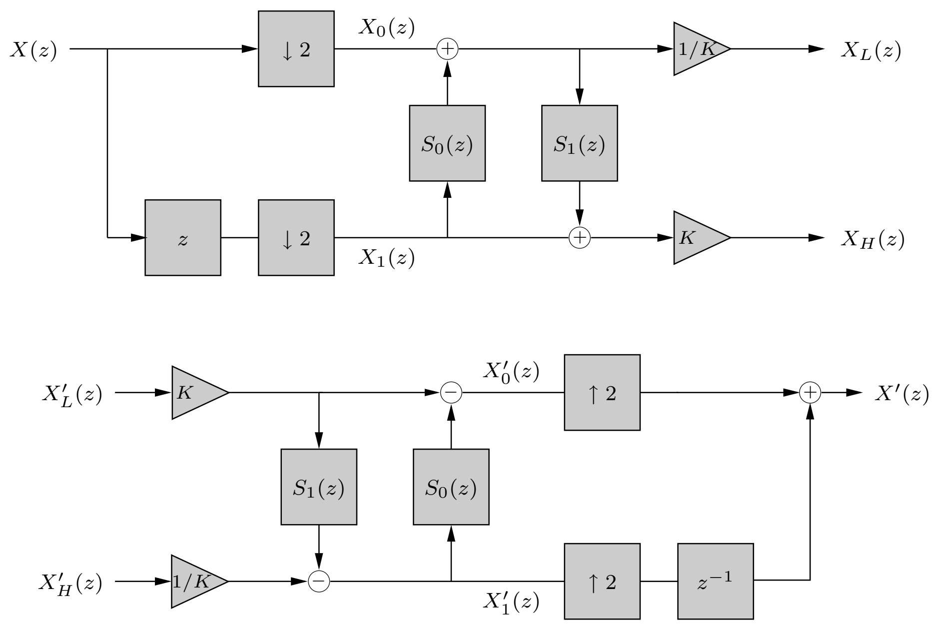

Figure 4. Lifting factorisation structure of a two-channel filter bank.

Table 1. Lifting factorisations for Haar, 2-6, 5-3, and 6-2 filter banks. The 5-3 filter bank is the filter bank selected for reversible coding in Part 1 of the JPEG2000 standard, and the 6-2 filter bank is the 2-6 filter bank with the analysis and synthesis stages switched.

	Haar	2-6	5-3	6-2
Number of steps	2	3	2	3
S_0 updates channel	Odd	Odd	Odd	Even
$S_0(z)$	$-z^0$	$-z^0$	$-\frac{1}{2}z^1 - \frac{1}{2}z^0$	z^0
$S_1(z)$	$\frac{1}{2}z^0$	$\frac{1}{2}z^0$	$\frac{1}{4}z^0 + \frac{1}{4}z^{-1}$	$-\frac{1}{2}z^0$
$S_2(z)$		$-\frac{1}{4}z^1 + \frac{1}{4}z^{-1}$		$-\frac{1}{4}z^1 + \frac{1}{4}z^{-1}$

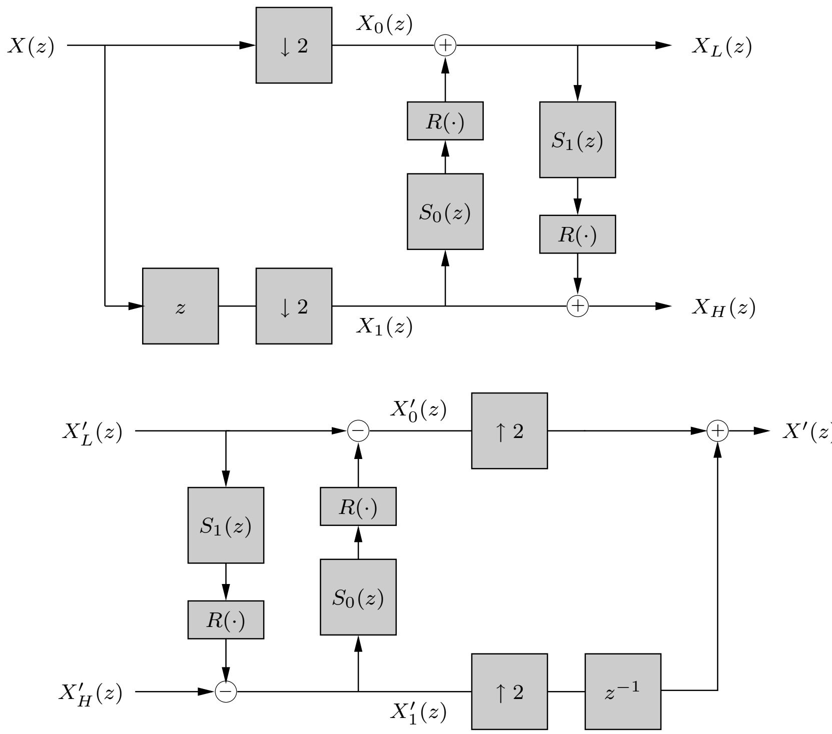

Figure 5. Lifting factorisation structure of a two-channel reversible filter bank.

4. SYMMETRIC EXTENSION

We have, thus far, only considered filter banks applied to inputs of infinite length — some form of input extension policy is necessary to deal with finite length inputs. Periodic extension, while simple, imposes restrictions on input lengths, and generates artifacts at signal boundaries. Symmetric extension of the input represents a more effective solution.[6] Different types of symmetric extension are possible since the point of symmetry may lie on a sample (whole sample symmetry) or between samples (half-sample symmetry), as illustrated for the $E_s^{(1,1)}$ and $E_s^{(2,2)}$ extensions in Figure 6 (notation follows that established in the paper[6] by Brislawn rather than that of the JPEG2000 standard).

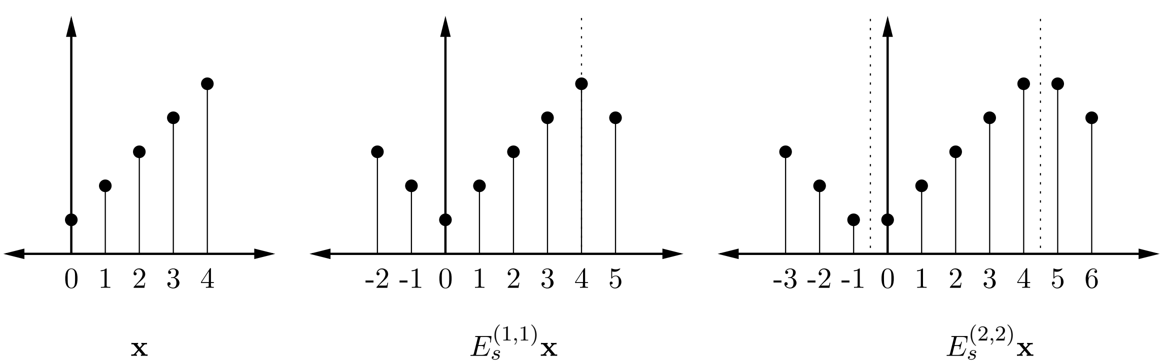

Figure 6. A signal and a single period of its $E_s^{(1,1)}$ and $E_s^{(2,2)}$ extensions.

Symmetric filter banks consist either of odd-length low- and highpass filters, both of which are symmetric, or even-length low- and highpass filters, one of which is symmetric and the other anti-symmetric[1] — the former will be referred to as a whole-sample symmetric (WS) filter bank, and the latter as a half-sample symmetric filter bank (HS).[3] When the input to a symmetric filter bank is symmetrically extended using the extension appropriate to the analysis filter bank symmetries, the low- and highpass channels also have symmetries which allow the (in-principle) infinite-length channels to be restricted to a single period, and then returned to infinite-length by appropriate symmetric extension prior to reconstruction by the synthesis filter bank. A symmetric extension filter bank is *non-expansive* if the number of samples retained on the lowpass and highpass channels equals the number of samples on the input prior to symmetric extension. WS filter banks require an $E_s^{(1,1)}$ input extension (an extension with whole-sample symmetry at each end), and generate symmetric lowpass and highpass output channels, while HS filter banks require an $E_s^{(2,2)}$ input extension (an extension with half-sample symmetry at each end), and generate lowpass and highpass output channels which are symmetric and anti-symmetric respectively.[6]

5. SYMMETRIC EXTENSION AND LIFTING

A symmetric extension WS filter bank requires symmetric lowpass and highpass channels in order to be invertible and non-expansive. This is the same symmetry found on the even and odd channels of the polyphase decomposition of a symmetrically extended input, so the required channel symmetry may be guaranteed by ensuring that it is preserved by each lifting step of a lifting implementation. Since convolution of a symmetric channel with a symmetric filter generates a symmetric result, this requirement is met by the standard lifting factorisations for the widely used WS filter banks,[3] which consist of appropriately centred symmetric lifting steps.

HS filter banks, in contrast, require an anti-symmetric highpass channel on the analysis output. Since the polyphase decomposition does not generate these symmetries, the usual structure of a lifting factorisation for an HS filter banks consists of a sequence of initialisation steps (asymmetric, or trivially symmetric such as the Haar filter bank), which generate the required channel symmetries, followed by a sequence of appropriately centred anti-symmetric lifting steps, which preserve these symmetries.[3]

6. SYMMETRIC EXTENSION INTEGER TRANSFORMS

As discussed above, perfect reconstruction of a symmetric extension filter bank is dependent on the symmetries of the output channels, which allow the truncation operation at the end of the analysis filter bank to be reversed by application of the appropriate symmetric extensions to the inputs of the synthesis filter bank. When constructing integer transforms by adding rounding operations as discussed in Section 3, perfect reconstruction requires that the rounding operations not break these symmetries.

WS filter banks require symmetric low and highpass channels, so that integer to integer versions are easily constructed by insertion of *any* rounding procedure, since any such procedure preserves the symmetry of the update branch to which it is applied. HS filter banks, in contrast, require an anti-symmetric highpass channel — an inappropriate rounding operation may not preserve this symmetry on the update branch to which it is applied.

The first restriction arising from this more complex structure is that the rounding operation applied to the anti-symmetric lifting steps must be an odd function so that it does not destroy anti-symmetry. A second, and more problematic, issue is the rounding procedure for the initial steps prior to the emergence of symmetry and anti-symmetry — while appropriate choices are available for the Haar initialisation, some initialisations do not admit *any* workable choice of rounding.

7. ROUNDING OF ANTI-SYMMETRIC LIFTING STEPS

In Figure 7, assume that step $S(z)$ represents the first anti-symmetric step subsequent to the non-symmetric initialisation which has generated symmetric even channel $X_0(z)$ and anti-symmetric odd channel $X_1(z)$. Since $U(z)$ has the appropriate anti-symmetry in the absence of the rounding operation $R(\cdot)$, updated odd channel $X_1(z) + R(U(z))$ is anti-symmetric if $R(\cdot)$ preserves the anti-symmetry of $U(z)$, requiring that $R(-x) = -R(x)$.

While the floor operation $R_\beta(x) = \lfloor x+\beta \rfloor$ from Part 1 of the JPEG2000 standard does not satisfy this constraint, it is satisfied by a rounding procedure which takes the integer part of its argument

$$R'_\beta(x) = \begin{cases} \lfloor x+\beta \rfloor & \text{if } x \geq 0 \\ \lceil x-\beta \rceil & \text{if } x < 0 \end{cases} \qquad 0 \leq \beta < 1.$$

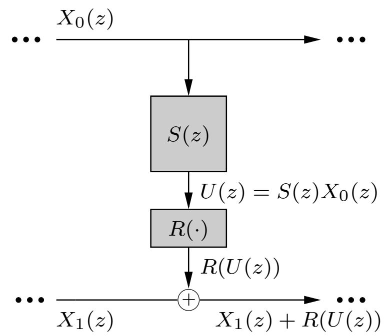

Figure 7. A single lifting step in an integer transform.

8. ROUNDING OF INITIALISATION STEPS

The more serious difficulty is demonstrated for the 6-2 filter bank of Table 1, illustrated in Figure 8. Imposing the constraint that the highpass channel be anti-symmetric at the end of the initialisation steps, we obtain

$$b + R\left(-\frac{a+b}{2}\right) = -\left(a + R\left(-\frac{a+b}{2}\right)\right) \quad \forall a, b \in \mathbb{Z},$$

which implies

$$\frac{x}{2} = R\left(\frac{x}{2}\right) \quad \forall x \in \mathbb{Z},$$

which is not a valid rounding procedure. There is no scalar rounding operation which preserves the required symmetry for this filter bank.

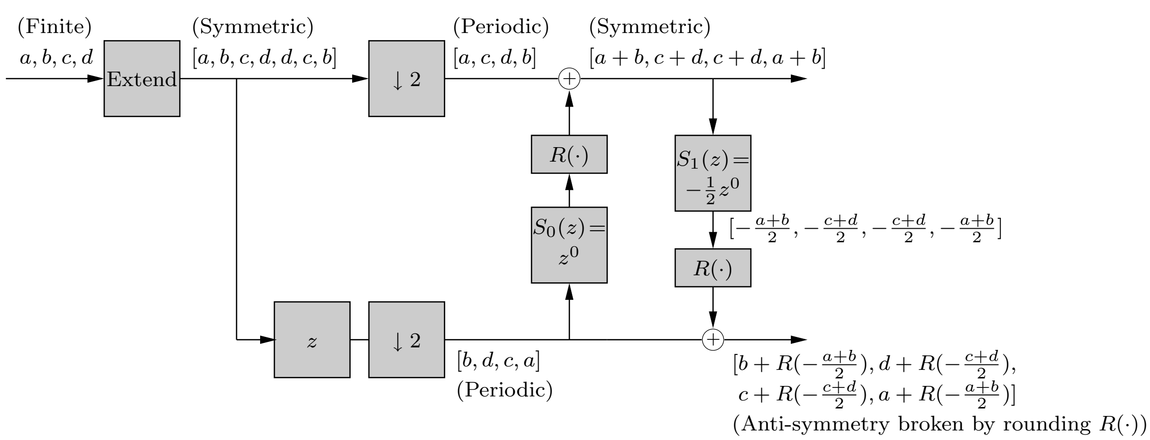

Figure 8. Non-symmetric initialisation steps of the 6-2 filter bank. A single period of an in-principle infinite signal is denoted by $[\ldots]$. The rounding operation on step $S_0(z)$ is irrelevant, since the step maps integers to integers even in the absence of rounding.

9. AN EXCEPTIONAL CLASS OF FILTER BANKS

As a result of an interesting coincidence, the well known 2-6 and 2-10 filter banks (as well as any others with factorisations consisting of two Haar initialisation steps followed by a single anti-symmetric step) are reversible with the usual floor rounding operation (this is probably a significant reason for the failure of reversibility described above not having been previously observed). These filters consist of two Haar initialisation steps followed by a single anti-symmetric step. While the correct symmetry and anti-symmetry is generated by the Haar initialisation using floor-based rounding, the same rounding destroys anti-symmetry on the highpass channel after the anti-symmetric step, as described above. However, since the Haar initialisation steps both consist of single filter taps at z^0, and they are followed by a single anti-symmetric odd-update step, the error resulting from extension of the non-symmetric highpass channel has no effect — it is irrelevant to the first synthesis step since it is an odd channel update, and has no effect on the Haar synthesis steps since they do not see the extended part of the even or odd channels (see Figure 9).

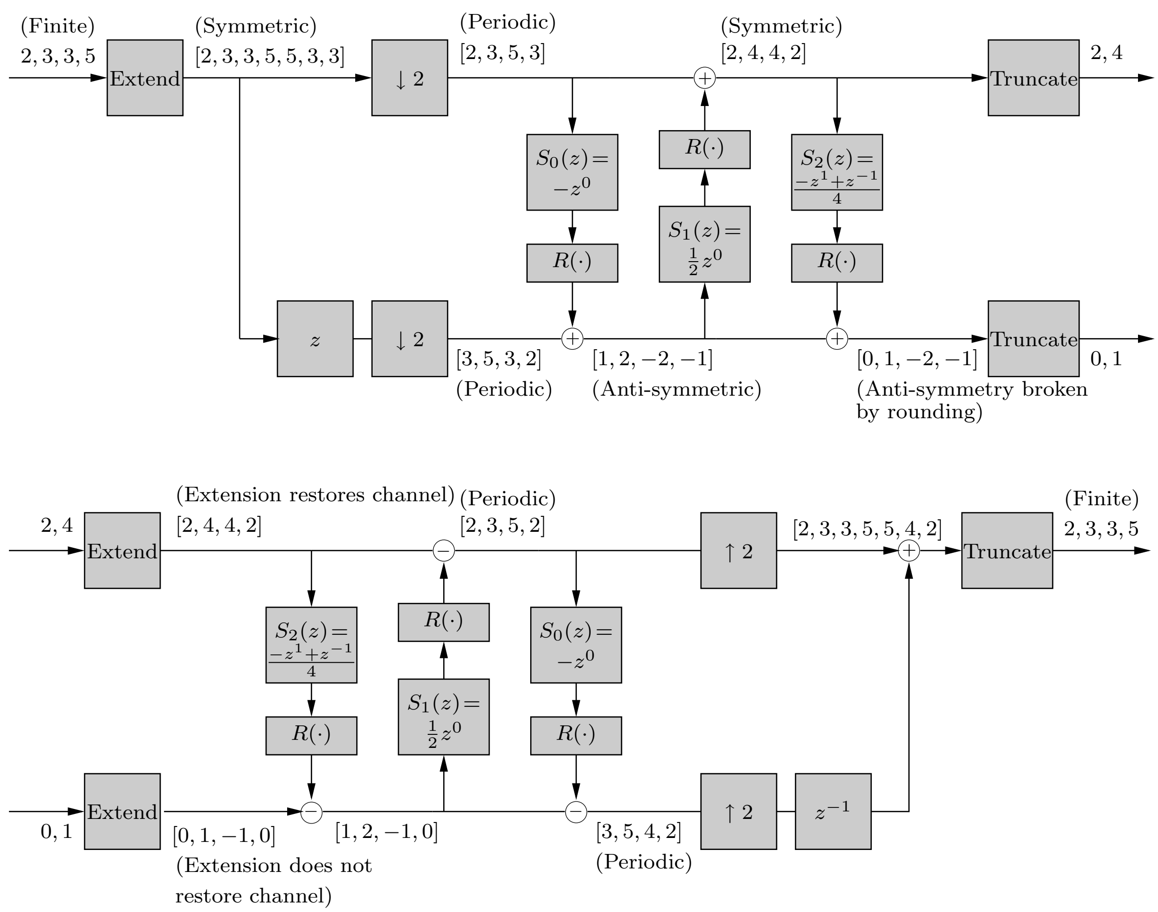

Figure 9. Reversible 2-6 filter bank. A single period of an in-principle infinite signal is denoted by $[\ldots]$. The rounding operation is $R(x) = \lfloor x \rfloor$. While synthesis does not correctly restore the full symmetric input, the original finite extent part of the input is correctly reconstructed because the first synthesis step is computed from the even channel, which is correct, and the subsequent steps are single taps at z^0 and therefore do not transfer information from the incorrect extension into the correct original part of the signal.

10. RANGE SPACE PRESERVING QUANTISATION

Since no scalar rounding operation suffices, we now examine a more general class of operations which, while impractical, does, in principle, enable the construction of reversible non-expansive HS filter banks with symmetric pre-extension. This discussion is most conveniently presented using a simple vector space representation in which a fixed length input vector in $\mathbb{R}^N$ is mapped to a single period of each of the even and odd channels after symmetric extension and polyphase decomposition, and the action of filters is represented by linear operators. Consider applying $E_s^{(2,2)}$ pre-extension and the polyphase decomposition to input vector $\mathbf{u} \in \mathbb{R}^N$. Retain a single period of each of the even and odd channels, labelling them $\mathbf{v}_0$ and $\mathbf{v}_1$ respectively — these vectors are both in $\mathbb{R}^N$, and $\mathbf{v}_1$ is a mirror image of $\mathbf{v}_0$. Now, define the operator $P_{HS} : \mathbb{R}^N \mapsto \mathbb{R}^{2N}$ which maps $\mathbf{u}$ to the block vector $\begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \end{pmatrix}$. Since the odd channel is a mirror image of the even channel, the range space (we label it $R_{HS,0}$) of this mapping is an N dimensional subspace of the $2N$ dimensional parent space $\mathbb{R}^{2N}$.

Within this representation, a lifting filter bank is expressed as $S_{L-1} \ldots S_1 S_0 P_{HS}$, where each lifting step is defined as $S_k = I + U_k$. The structure of the U_k

$$U_k = \begin{pmatrix} 0 & F_k \\ 0 & 0 \end{pmatrix} \text{ even update} \qquad U_k = \begin{pmatrix} 0 & 0 \\ F_k & 0 \end{pmatrix} \text{ odd update}$$

depends on whether the step is an even or an odd update. Define $R_{HS,1}$ as the N dimensional subspace of $\mathbb{R}^{2N}$ within which each vector has a symmetric upper component and an anti-symmetric lower component. When the filter bank is HS, non-expansiveness and invertibility depend on the the range space of the entire filter bank $S_{L-1} \ldots S_1 S_0 P_{HS}$ having the symmetry properties necessary for being a subspace of $R_{HS,1}$. These symmetry properties emerge progressively as each of the initialisation lifting steps maps its input to a new range space. The range space after the final initialisation step (that is, the last step before the anti-symmetric steps that preserve the channel symmetries) is a subspace of $R_{HS,1}$.

A reversible filter bank is constructed by replacing each step S_k with $S'_k = I + Q_k U_k$, where Q_k is a rounding operation for step k. Since the required channel symmetries only emerge after the initialisation steps, simple scalar rounding for these steps will, in general, alter the effective range space after each S'_k, so that their product will no longer have a range space which is a subspace of $R_{HS,1}$. (This problem does not arise in WS filter banks since the required symmetry is present directly after the polyphase decomposition, and is preserved by each symmetric lifting step — while the insertion of a scalar rounding does not preserve the specific range space of the modified step, the range space remains a subspace of the appropriate symmetric parent space.) A sufficient condition for the final symmetries to be preserved is that each rounding operation Q_k not remove its operand from the range space of $U_k S_{k-1} \ldots S_1 S_0 P_{HS}$ — in general this requires a vector rounding operation (i.e. Lattice Vector Quantisation), such as

$$Q_k(\mathbf{u}) = \underset{\mathbf{v} \in \mathbb{Z}^{2N}, \mathbf{v} \in \operatorname{ran}(U_k S_{k-1} \ldots S_1 S_0 P_{HS})}{\arg\min} \|\mathbf{u} - \mathbf{v}\| .$$

In addition to having greater computational requirements than scalar rounding, use of such an operation is likely to result in a filter bank with poor performance, representing a poor approximation to the corresponding irreversible filter bank, since the vector rounding operation quantises to the lattice which is the intersection of the appropriate integer lattice and the relevant lifting step range space — this lattice may be much less dense than the integer lattice, and therefore have a much greater rounding error.

11. CONCLUSIONS

There appears to be no practical choice of rounding rules which would guarantee reversibility for all HS integer transform filter banks based on symmetric pre-extension. An alternative *iterated* extension approach,[3] in which extension is performed directly prior to convolution in each lifting step, has been included in Part 2 of the JPEG2000 standard. This algorithm is equivalent to symmetric pre-extension for WS filter banks, but not for HS filter banks.

REFERENCES

1. G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA, USA, 1996.
2. P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.

3. C. Brislawn, B. Wohlberg, and A. Percus, “Resolution scalability for arbitrary wavelet transforms in the JPEG-2000 standard,” in *Visual Communications and Image Processing (VCIP) 2003*, (Lugano, Switzerland), July 2003.
4. I. Daubechies and W. Sweldens, “Factoring wavelet transforms into lifting steps,” *Journal Of Fourier Analysis And Applications* **4**(3), pp. 247–269, 1998.
5. A. R. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo, “Wavelet transforms that map integers to integers,” *Applied and Computational Harmonic Analysis* **5**, pp. 332–369, July 1998.
6. C. M. Brislawn, “Classification of nonexpansive symmetric extension transforms for multirate filter banks,” *Applied and Computational Harmonic Analysis* **3**, pp. 337–357, 1996.